2017

GAMS Handout 1

Ethan Yang
yey217@lehigh.edu

The General Algebraic Modeling System

Y. C. Ethan Yang
Department of Civil and Environmental Engineering
Lehigh University
Bethlehem, PA USA

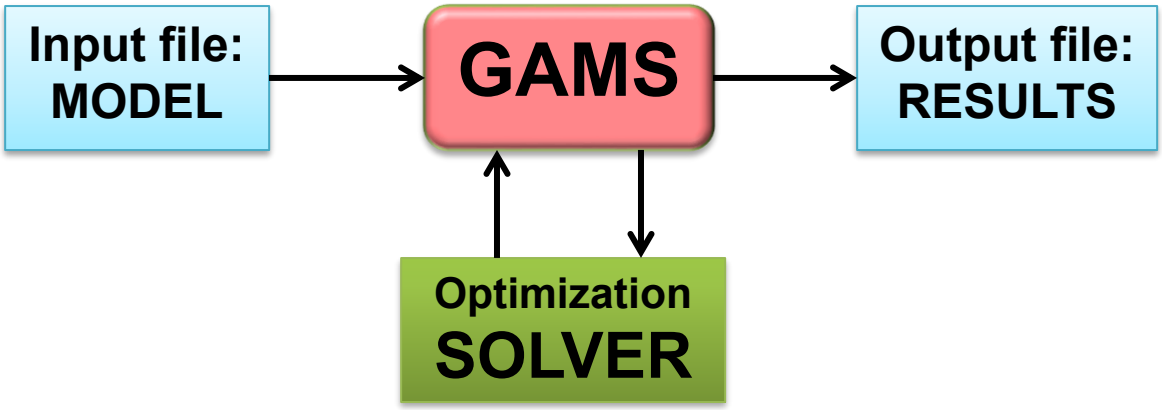
YOU WILL LEARN

- What is General Algebraic Modeling System (GAMS)?
- The installation of GAMS
- How to navigate the GAMS interface?
- The basic components of GAMS input file
- How to read GAMS output file?
- How to organize GAMS results?

} Through an example of water resources management

WHAT IS GAMS?

- The General Algebraic Modeling System (GAMS) is a high-level modeling system for optimization. GAMS is designed to make the construction and solution of large and complex mathematical programming modeling.



- Roots: World Bank, 1976
- Went commercial in 1987: GAMS Development Corp.
- Broad academic & commercial user community and network
- Detailed Information in:
  - GAMS Guide and Tutorials

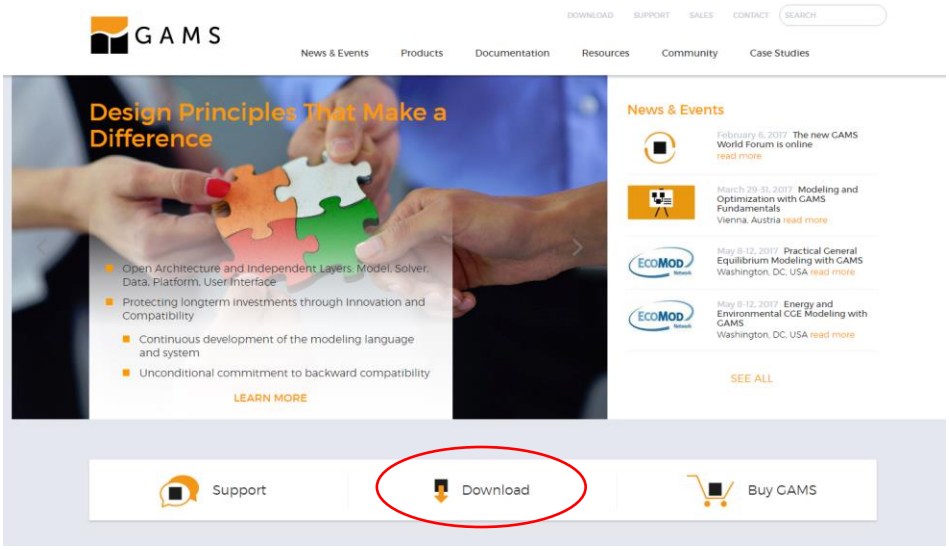
Rosenthal, R. E. (2010). GAMS – A User's Guide. GAMS Development Corporation. Washington D.C.

- GAMS website

  Official website: www.gams.com
  Support wiki: http://support.gams-software.com
  Interface wiki: http://interfaces.gams-software.com
  Google Group: http://groups.google.com/group/gamsworld

THE INSTALLATION OF GAMS

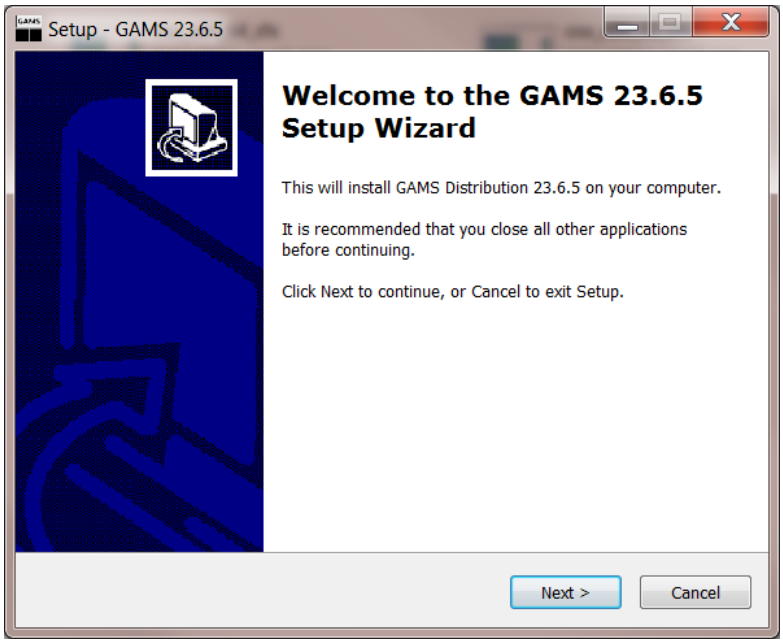
- Go to GAMS website at www.gams.com and download the free version



- GAMS works on all operation systems: Windows, Unix and Wine.

24.5.6	September 23, 2015	AIX | Linux 64 bit | WINE (Linux) | Mac OS X Intel 64 bit | Solaris SPARC 64 bit | Solaris x64 64 bit | Windows 64 bit | Windows 32 bit
24.4.6	December 20, 2014	AIX | Linux 64 bit | WINE (Linux) | Mac OS X Intel 64 bit | Solaris SPARC 64 bit | Solaris x64 64 bit | Windows 64 bit | Windows 32 bit
24.3.3	July 31, 2014	AIX | Linux 64 bit | WINE (Linux) | Mac OS X Intel 64 bit | Solaris SPARC 64 bit | Solaris x64 64 bit | Windows 64 bit | Windows 32 bit
24.2.3	December 9, 2013	AIX | Linux 64 bit | Linux 32 bit | WINE (Linux) | Mac OS X Intel 64 bit | Solaris SPARC 32 bit | Solaris SPARC 64 bit | Solaris x64 64 bit | Windows 64 bit | Windows 32 bit
24.1.3	May 30, 2013	AIX | Linux 64 bit | Linux 32 bit | WINE (Linux) | Mac OS X Intel 64 bit | Solaris SPARC 32 bit | Solaris SPARC 64 bit | Solaris x64 64 bit | Windows 64 bit | Windows 32 bit
24.0.2	December 24, 2012	AIX | Linux 64 bit | Linux 32 bit | WINE (Linux) | Mac OS X Intel 64 bit | Solaris SPARC 32 bit | Solaris SPARC 64 bit | Solaris x64 64 bit | Windows 64 bit | Windows 32 bit
23.9.5	July 04, 2012	AIX | Linux 64 bit | Linux 32 bit | WINE (Linux) | Mac OS X Intel 64 bit | Mac OS X Intel 32 bit | Solaris SPARC 32 bit | Solaris SPARC 64 bit | Solaris x64 64 bit | Windows 64 bit | Windows 32 bit
23.8.2	March 17, 2012	AIX | Linux 64 bit | Linux 32 bit | WINE (Linux) | Mac OS X Intel 64 bit | Mac OS X Intel 32 bit | Solaris SPARC 32 bit | Solaris SPARC 64 bit | Solaris x64 64 bit | Windows 64 bit | Windows 32 bit
23.7.3	July 14, 2011	AIX | Linux 64 bit | Linux 32 bit | WINE (Linux) | Mac OS X Intel 64 bit | Mac OS X Intel 32 bit | Solaris SPARC 32 bit | Solaris SPARC 64 bit | Solaris x64 64 bit | Windows 64 bit | Windows 32 bit
23.6.5	December 13, 2010	AIX | Linux 64 bit | Linux 32 bit | Mac OS X Intel 64 bit | Mac OS X Intel 32 bit | Solaris SPARC 32 bit | Solaris SPARC 64 bit | Solaris x64 64 bit | Windows 64 bit | Windows 32 bit
23.5.2	July 4, 2010	AIX | Linux 64 bit | Linux 32 bit | Mac OS X Intel 64 bit | Mac OS X Intel 32 bit | Solaris SPARC 32 bit | Solaris SPARC 64 bit | Solaris x64 64 bit | Windows 64 bit | Windows 32 bit
23.4.3	May 24, 2010	AIX | Linux 64 bit | Linux 32 bit | Mac OS X Intel 64 bit | Mac OS X Intel 32 bit | Solaris SPARC 32 bit | Solaris SPARC 64 bit | Solaris x64 64 bit | Windows 64 bit | Windows 32 bit
23.3.3	November 1, 2009	Linux 64 bit | Linux 32 bit | Mac OS X PPC | Mac OS X Intel 32 bit | Mac OS X Intel 32 bit | Solaris SPARC 32 bit | Solaris SPARC 64 bit | Solaris x64 64 bit | Windows 64 bit | Windows 32 bit

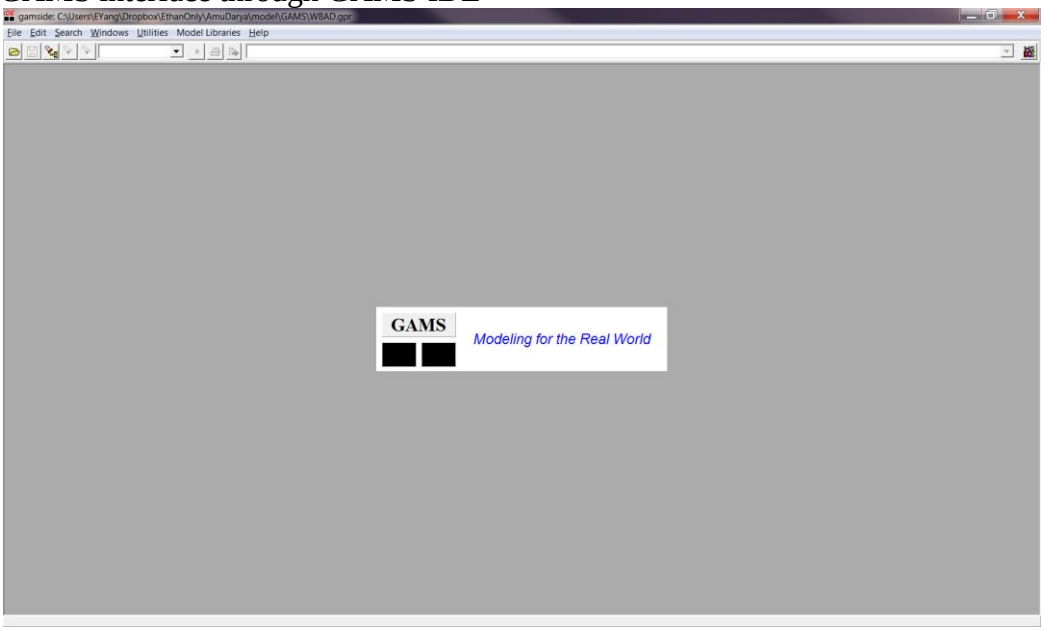
- Download the appropriate one and here is the example of install the downloaded file in Windows OS.



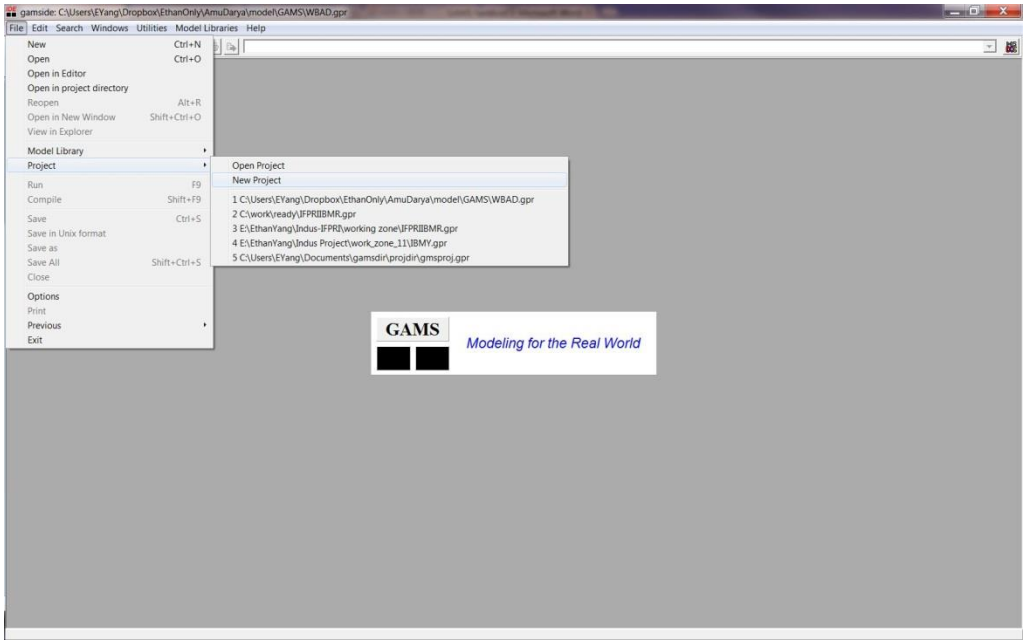
- The installation includes GAMS IDE (Integrated development environment) that allows users to activate the GAMS interface.

HOW TO NAVIGATE THE GAMS INTERFACE?

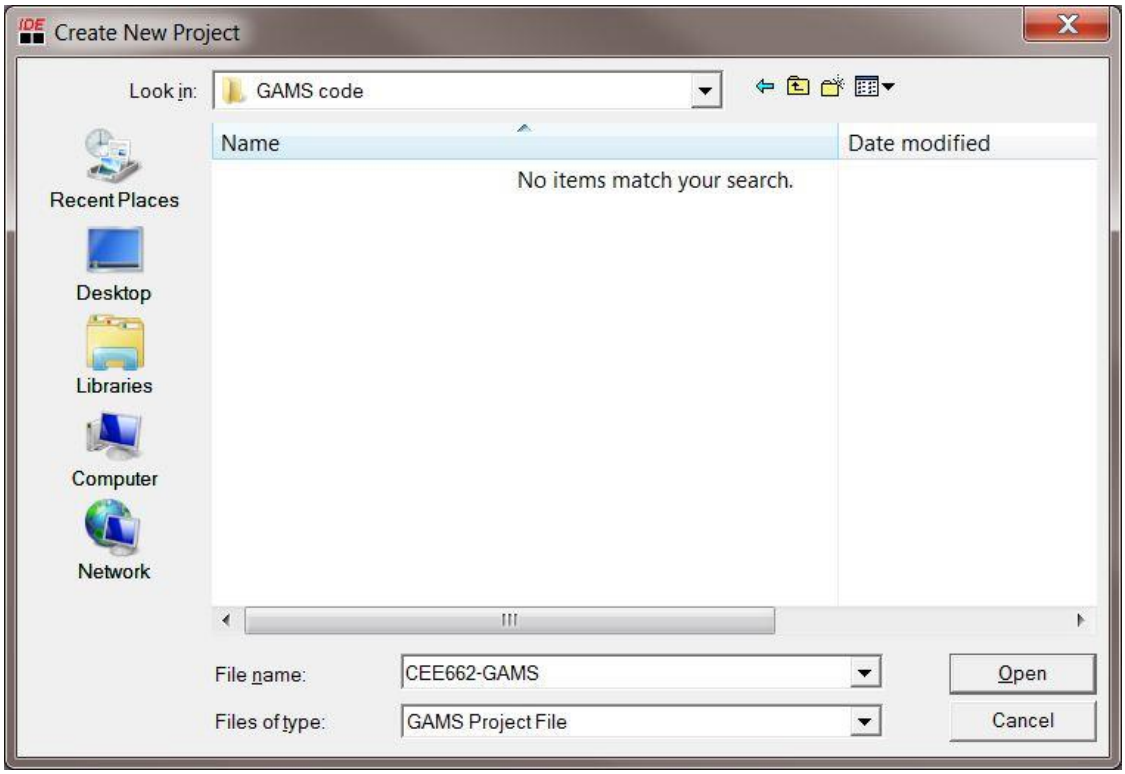
- Open GAMS interface through GAMS IDE



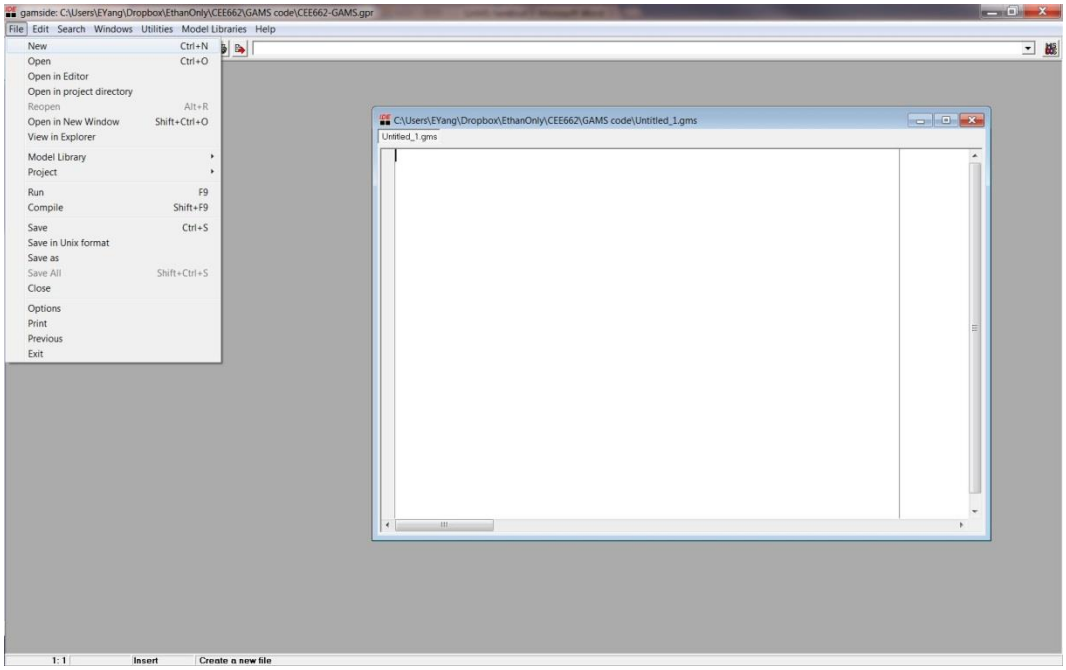
- First, you need to create a "project" file before you can do anything. Go to "**File**" tab in GAMS interface, select "**Project**" from the drop down menu and choose "**New Project**."



- Name your project with an appropriate name and save it into a folder



- Next, go to "**File**" tab in GAMS interface, select "**New**" from the drop down menu and a blank GAMS input file (.gms) will be created.



- Now, you are ready to code in GAMS!

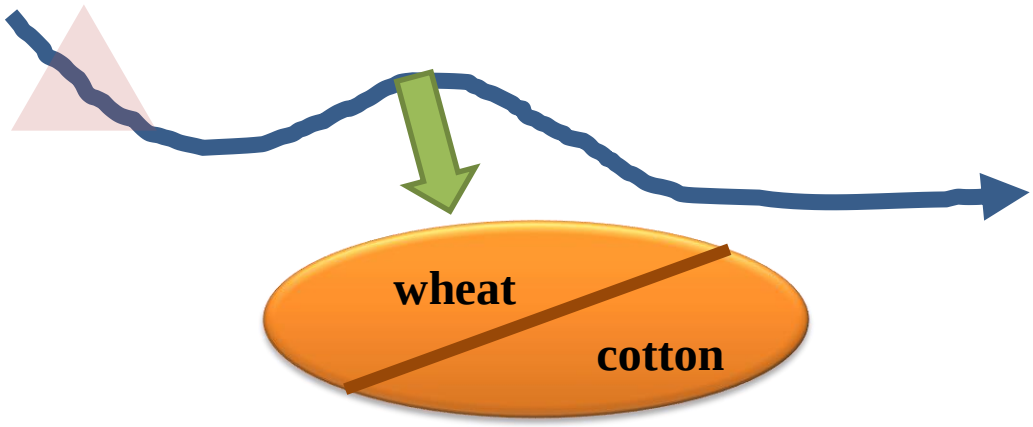
THE BASIC COMPONENTS OF GAMS INPUT FILE

- The GAMS input file is in general organized into the following sections
  - Specification of indices (**Sets**)
  - Specification of data (**Parameters/Tables**)
  - Listing of names and types of variables (**Decision variables and Objectives**)
  - Definition of the equations (**Constraints and Objective function**)
  - Specification of **bounds**, initial values and special options
  - Call to the optimization **solver**

- Some tips before constructing a GAMS input file
  - Terminate every statement with a **semicolon**
  - Case **insensitive**
  - Using the symbol "*****" to add **notation** to explain your code

- The creation of GAMS entities involves two steps:
  - **Declaration** means declaring the existence of something and giving it a name
  - **Assignment** or **definition** means giving something a specific value or form

- Example setting
  Let's say you are a farmer that wants to decide what crop you want to grow in your farm to **maximize** your profit. You have two different chooses: **wheat** and **cotton**. They require different amount of resources (**water** and **land**) and give you different net profit. All data you need are given in the following tables.



Net profit	USD
wheat	2
cotton	3

Resources need	water	land
wheat	2	3
cotton	4	2

Total available resources	
water	20
land	25

- **Let's play the game with GAMS!**
  - Define **Set**
    We need two different sets in this case: crops and resources.

```
Set c crop / wheat, cotton /;
Set input input need for crop / water, land /;
```

  - Define **Data**(Parameter, Table and Scalar)
    Using Parameter, Table and Scalar to input the data needed for the question.

```
Parameter Profit(c) net profit per unit production /
wheat  2
cotton 3/;

Table Res(c, input) resources need per unit production
        water  land
wheat     2     3
cotton    4     2
;

Scalar TWater total available water /20/;
Scalar TLand  total available land /25/;
```

  - Define **Variable**
    Two variables are needed: objective value and decision variable

```
Variable
Z       total profit
X(c)    production of crop
;
```

  - Define **Equation**
    Three equations are needed: objective function, constraint of water and constraint of land.

Equation
Obj Objective function: net profit from crop production
ConW Water constraint
ConL Land constraint
;

Obj..
Z =e= SUM(c,Profit(c)*X(c));

淨收入 = 麥產量 X 麥價格 + 棉花產量 X 棉花價格
= SUM(作物，作物價格 X 作物產量)

ConW..
SUM(c, X(c)* Res(c, "water")) =L= Twater;
麥的總用水量+棉花的總用水量 < 總用水量
麥單位用水量 X 麥產量+棉花單位用水量 X 棉花產量< 總用水量
SUM(作物, 作物產量 X 作物單位用水量) < 總用水量

ConL..
SUM(c, X(c)* Res(c, "land")) =L= Tland;
SUM(作物, 作物產量 X 作物單位用土地量) < 總用土地量

X.LO(c) = 0;
麥產量>=0
棉花產量>=0
作物產量>=0

- Define **Model**
  Give the model we just built a name

MODEL ALLEqu /OBJ, ConW, ConL/
MODEL OnlyWater /OBJ, ConW/

MODEL OnlyLand /OBJ, ConL/

- Define **Solver**
  Tell GAMS to solve the model using what method and driven by what objective.

  SOLVE GAMSdemo MAXIMIZING Z USING LP;

HOW TO READ GAMS OUTPUT FILE?

- Standard GAMS output includes
  - **Echo Print**
    Whether or not errors prevent your optimization problem from being solved, the first section of output from a GAMS run is an echo, or copy, of your input file.

    For the sake of future reference, GAMS puts line numbers on the left-hand side of the echo.

  - **Error Message**
    When the GAMS compiler encounters an error in the input file, it inserts a coded error message inside the echo print on the line immediately following the scene of the offense.

    These messages always start with **** and contain a '$' directly below the point at which the compiler thinks the error occurred.

    The $ is followed by a numerical error code, which is explained after the echo print. Several examples follow.
    It is not uncommon for one little offense like our missing semicolon to generate five intimidating error messages.

    The lesson here is: concentrate on fixing the first error and ignore the other!

    Unfortunately, you cannot always expect error messages to be so accurate in their advice. The compiler cannot read your mind.

  - **Equation Listings**
    A product of the solve command, the equation listing shows the specific instance of the model that is created when the current values of the sets and parameters are plugged into the general algebraic form of the model.

    The default output also contains a section called the column listing, analogous to the equation listing, which shows the coefficients of three specific variables for each generic variable.

- **Model Statistics**
  A group of statistics about the model's size.

  The BLOCK counts refer to the number of generic equations and variables. The SINGLE counts refer to individual rows and columns in the specific model instance being generated.

  For nonlinear models, some other statistics are given to describe the degree of non-linearity in the problem.

- **Status Reports**
  After the solver executes, GAMS prints out a brief solve summary whose two most important entries are SOLVER STATUS and the MODEL STATUS.

  The status reports are preceded by the same **** string as an error message.

  SOLVER STATUS/ MODEL STATUS
  1 NORMAL COMPLETION
  2 LOCALLY OPTIMAL
  3 UNBOUNDED
  4 INFEASIBLE

- **Solution Reports**
  If the solver status and model status are acceptable, then you will be interested in examining the results of the optimization.

  Standard mathematical programming output format.

  In this format, there is a line of printout for each row and column giving the lower limit, level, upper limit, and marginal.

HOW TO ORGANIZE GAMS RESULTS?

- The easiest way to show the out you want is through "DISPLAY" comment at the end of input file

  DISPLAY X.l, Z.l;

- GAMS has a function that help users to organize output. The syntax "GDX" is a useful one to summarized ALL INPUTS and RESULTS from your model. We will see how "GDX" works for the exercises in the rest of the course.